Reactor & Webflux

Reactive Streams for the JVM

Problem: React to asynchronous events

- Async reactions to events
- Today done with callbacks
- But what happens when the reactions requires another callback?
  - … and this reaction still another callback
    - … and still another one??

- One of the main goals of this is to address the problem of back pressure.
  - Back pressure means that our consumer should be able to tell the producer how much data he should send.

Callback Hell - Pyramid of Doom

```
func1(param, function(err, res) {
  func2(param, function(err, res) {
    func3(param, function(err, res) {
      func4(param, function(err, res) {
        func5(param, function(err, res) {
          func6(param, function(err, res) {
            func7(param, function(err, res) {
              func8(param, function(err, res) {
                func9(param, function(err, res) {
        // Do something...
          });
        });
      });
    });
        });
  });
  });
});
```

Solution proposal: Reactor

- Has its roots in .NET
- Implements the concept of the **Reactive Streams Specification**
- Basic elements are Mono and Flux

Observables – asynchronous iterators

Synchron

Single value

```
val value: T
```

Multiple values

```
val values: Iterator<T>
```

Asynchron

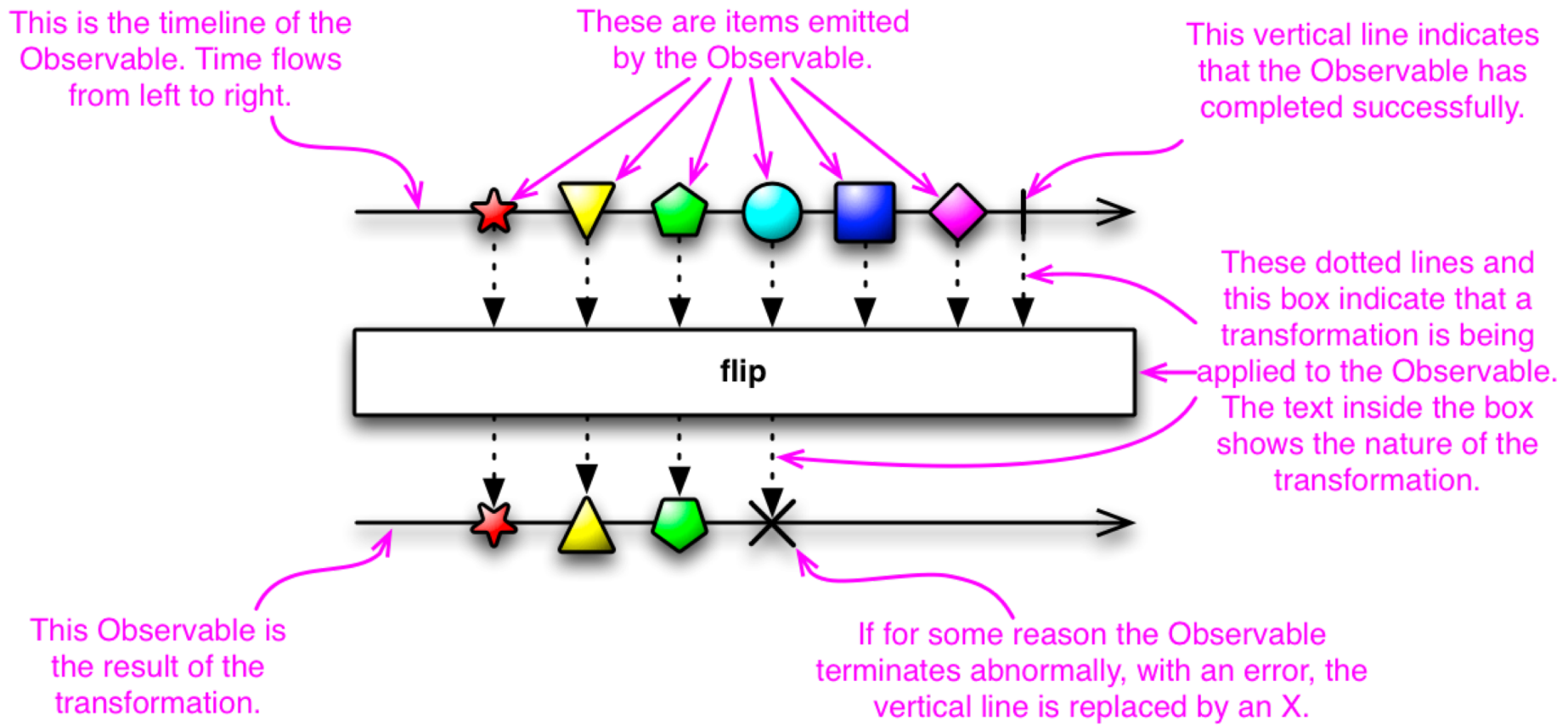
Single, asynchronous value

```
val futureValue: Future<T>
```

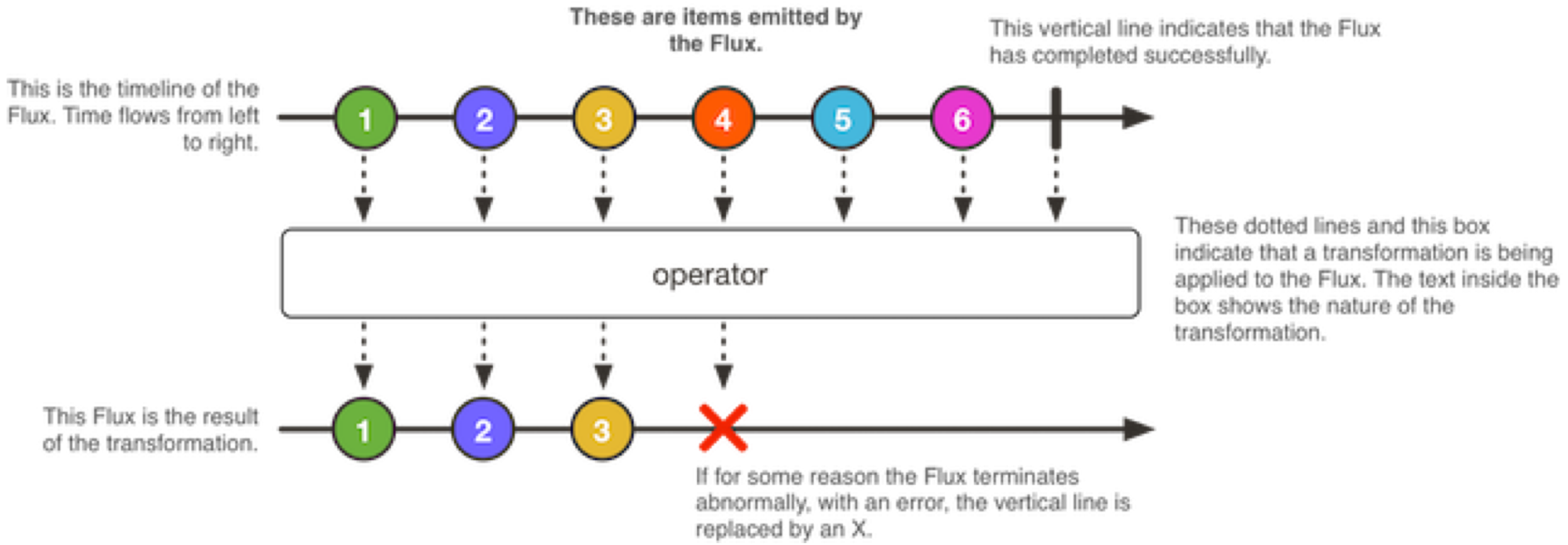
Multiple, asynchronous values

```
val futureValues: Observable<T>
```

Illustration of events streams – Marble Diagramme

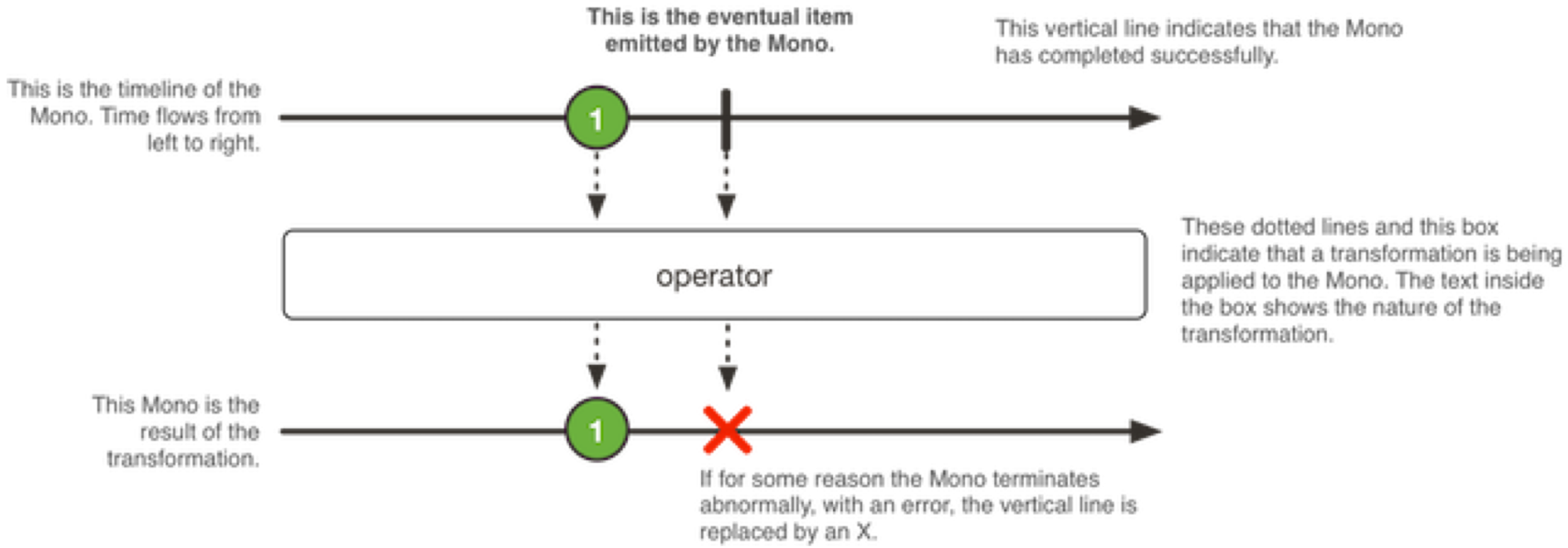


Flux – Asynchronous sequence of 0-N items



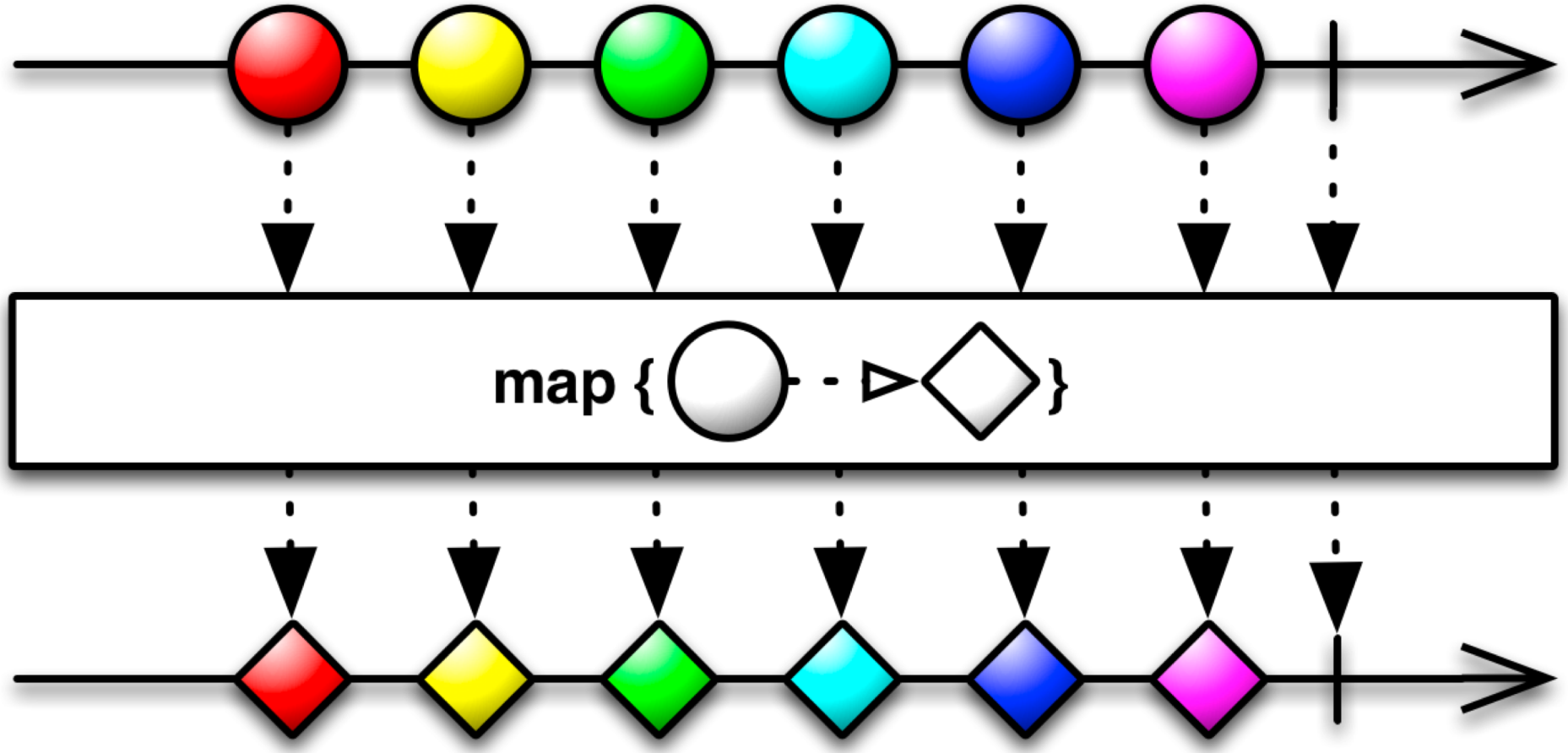
https://projectreactor.io/docs/core/release/reference/docs/index.html#getting-started-introducing-reactor

Mono – Asynchronous 0-1 result

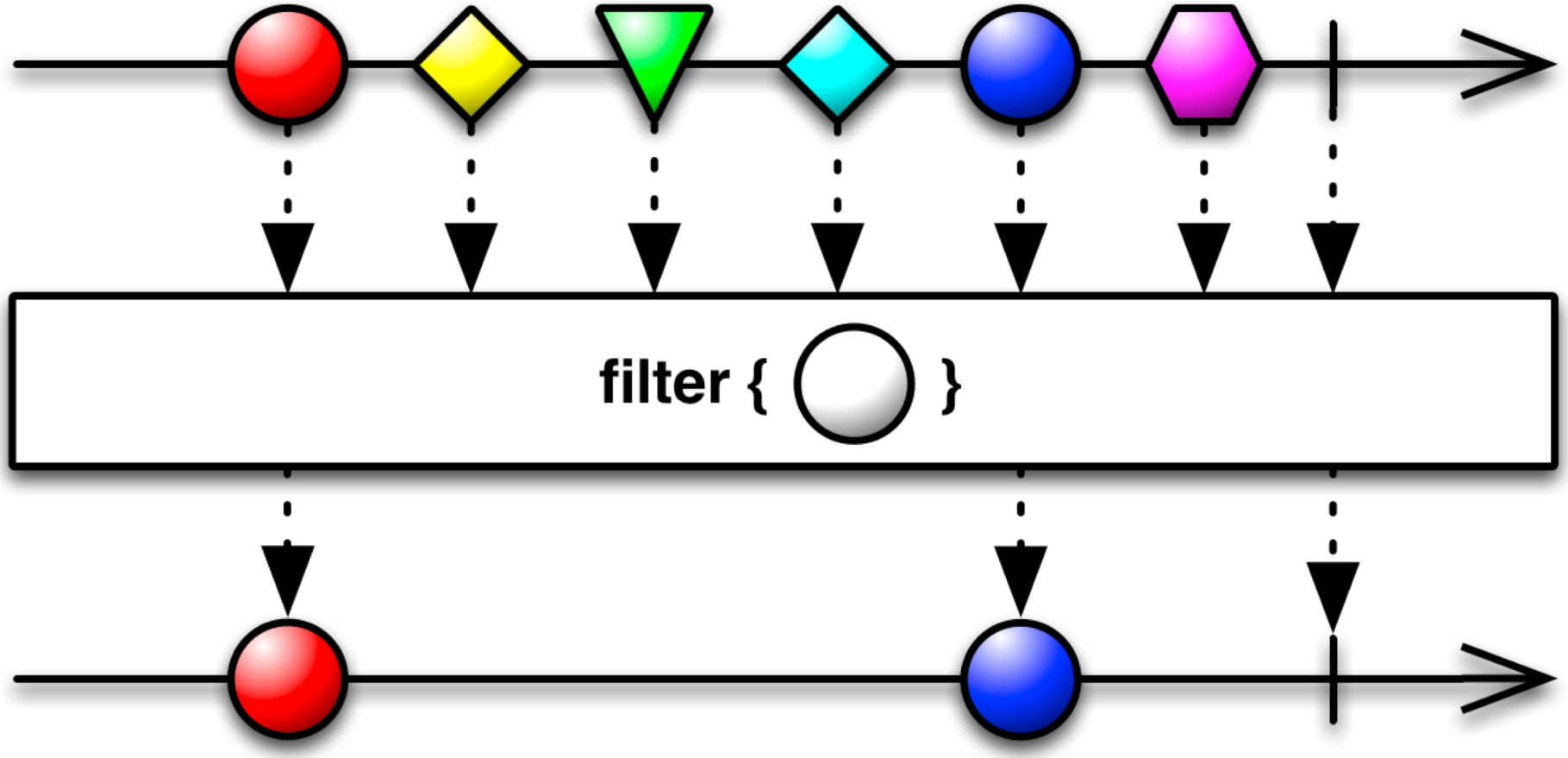


https://projectreactor.io/docs/core/release/reference/docs/index.html#getting-started-introducing-reactor

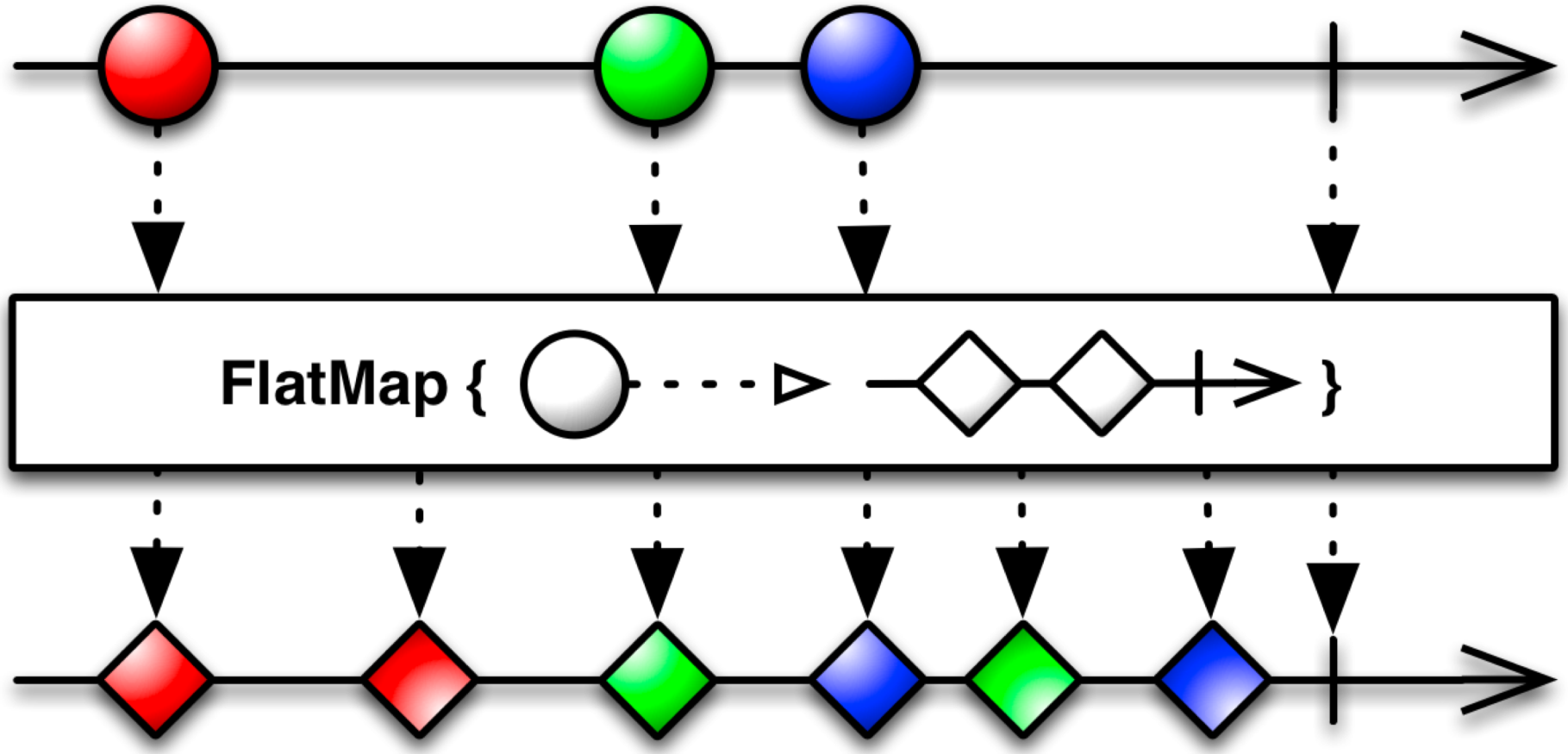
Map



Filter



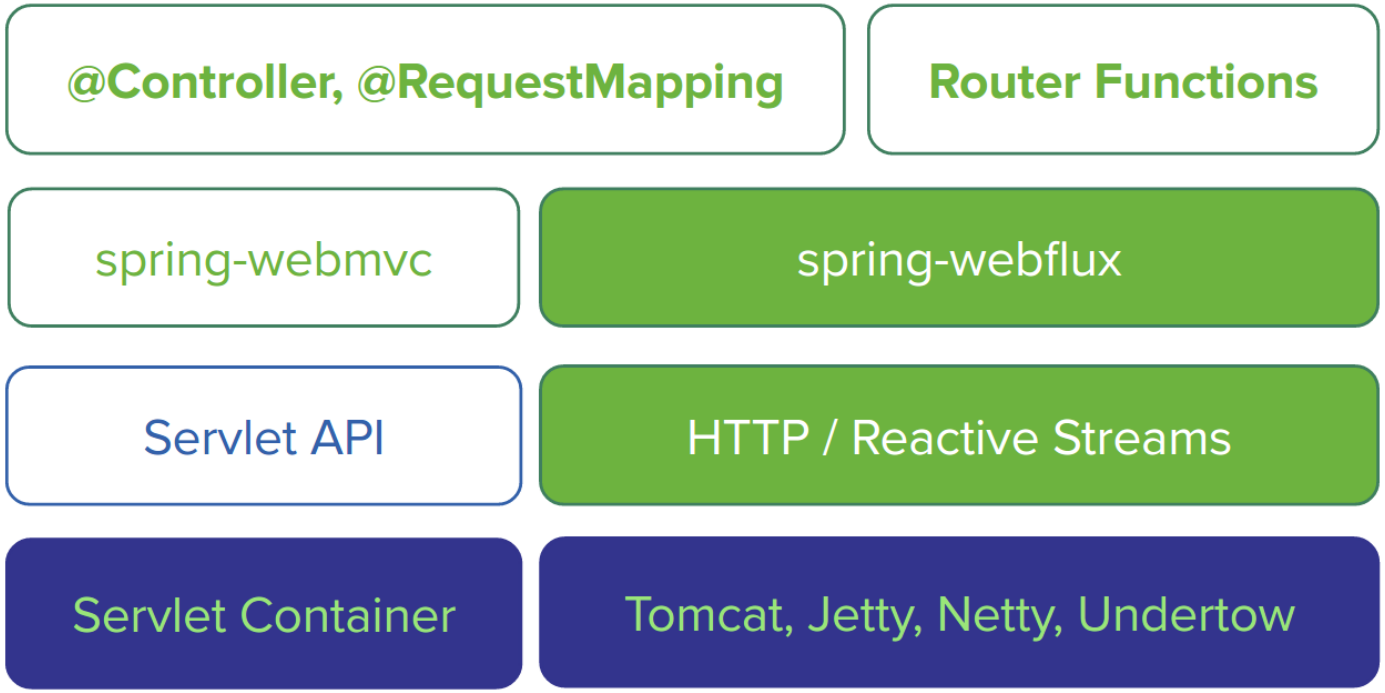
FlatMap



Remember **NOTHING** happens if you dont

SUBSCRIBE

Webflux



https://docs.spring.io/spring/docs/5.0.0.BUILD-SNAPSHOT/spring-framework-reference/html/web-reactive.html